**Отчёт о проделанной работе над курсовым проектом**

**Техническое задание** для проекта по созданию *rest-service* имела следующую структуру:

1.Разработка REST-сервиса Кофейни "Verwee".

2.Описание схемы данных :

В схеме должно быть не менее 3-х таблиц.

услуги, предоставляемые кофейней:

продажа кофе и десертов.

таблицы:

покупатель Buyers(id,Name, Sale,id\_menu,id\_barista)

товары Goods(id,name,coasts,quantity,ingridients)

бариста Barista(id,FIO,job tittle,expiriens,id\_menu,id\_goods)

меню Menu(id,id\_goods,position name)

Связи таблиц:

Barista1-N Bayers;

Barista 1-N Goods;

Menu 1-N Goods;

Menu 1-N Buyers.

3.CRUD операции к каждой таблице;

2 эндпоинта:

1. с вычисляемым полем

2. с данными из нескольких таблиц

Эндпоинты должны возвращать только указанные поля данных

1)для получения данных о количестве N-ой позиции в меню: название позиции,количество этой позиции.

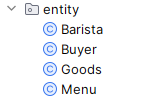
2)для получения данных о количестве наиболее популярной позиции у определенного покупателя: Имя Клиента, количество заказанных позиций.

**Реализация Rest-service**

1. Для создания REST-сервиса с использованием Spring Framework сначала создаем новый проект Maven. Затем добавляем необходимые зависимости для работы с REST.



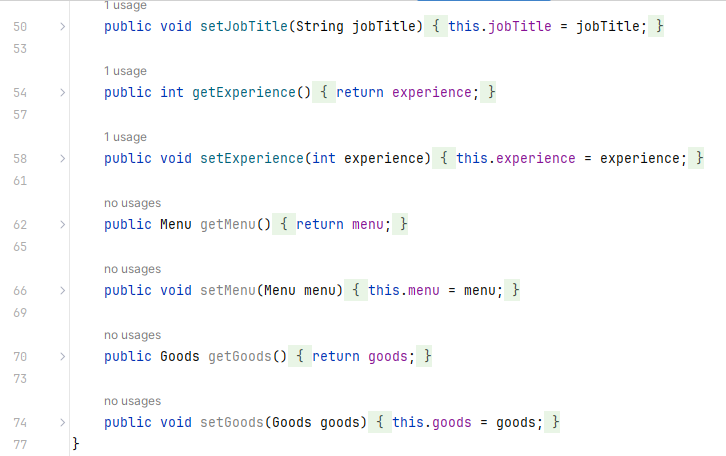
1. "Определение моделей данных" в разработке REST-сервиса.



Из ТЗ был определен набор сущностей, необходимых для проекта, в которых были определены поля и методы.







Также аналогично были созданы и остальные сущности.

3.Создание **Контроллеров**

Контроллеры в Spring обрабатывают HTTP-запросы, маршрутизируют их к соответствующим методам приложения, извлекают и обрабатывают данные запроса, формируют HTTP-ответы и управляют потоком выполнения запроса.

Реализация контроллеров была произведена в одном общем классе(в последствие можно будет для удобства тестирования разделить все на разные классы)





Этот контроллер является точкой входа для обработки запросов, связанных с кофейней. Он использует репозитории для доступа к данным о покупателях, товарам, баристах и меню.





Эти методы контроллера обрабатывают HTTP-запросы для выполнения операций CRUD (create, read, update, delete) с данными о покупателях.

Так же аналогично были сделаны контроллеры goods, baristas, menu.

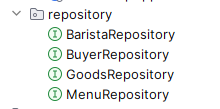
Данные REST не выполняет действий с БД либо другими WS он просто имитирует работу REST.

Значит в результате у нас будет одна ссылка:

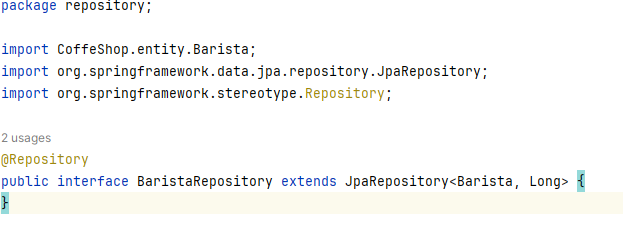
http://localhost:8080/(название контроллера (например, "/ menu")**)** – обращаясь по этой ссылке разными HTTP методами можно выполнять разные действия.

4.Создание **интерфейсов**

Папка repository в рест-сервисе отвечает за доступ к данным в базе данных через определенные интерфейсы, предоставляя методы для выполнения операций CRUD (create, read, update, delete) с объектами-сущностями приложения.



Структура реализованный интерфейсов.



Пример реализации одного из интерфейсов.

В этом интерфейсе репозитория (BaristaRepository) предоставляются базовые методы доступа к данным для сущности Barista. Он расширяет интерфейс JpaRepository, что позволяет использовать стандартные операции доступа к данным для сущности Barista. Также определен дополнительный метод, который может быть использован для выполнения пользовательских операций с данными.

Далее будут добавлены новые доработки, которые помогут сделать репозиторий более функциональным, производительным и гибким.

Аналогично были реализованы и другие интерфейсы.

На текущий момент выполнена основная часть работы, но впереди еще много задач по оптимизации, тестированию и улучшению REST-сервиса.